Choosing a Deployment Model

Eric Wright
Technology Evangelist, Podcaster

@DiscoPosse www.discoposse.com www.discopossepodcast.com

Overview

- What makes up a Nomad environment?
- Compare deployment examples
- Explore the Wired Brain Coffee use-case
- Map out a hybrid Nomad deployment
- See Nomad example using Infrastructure-as-Code with Terraform

Starter Course

Getting Started with HashiCorp Nomad

Eric Wright

Nomad Components and Dependencies

Nomad Dependencies

Ideal to have Consul as an auto-discovery + service mesh

Task drivers to support various jobs/workloads

Persistent storage plugins needed for stateful workloads

Secret management (e.g. Vault)

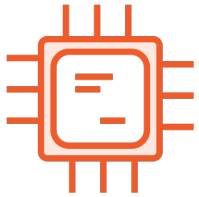
Monitoring for Day 2 operations

Compute Resources

Networking Resources

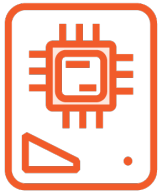
Storage Resources

Recommended Resource Node Minimums



4-8+ cores

16-32+ GB of RAM



40-80 GB of “fast” disk

Persistent Storage for Nomad Workloads

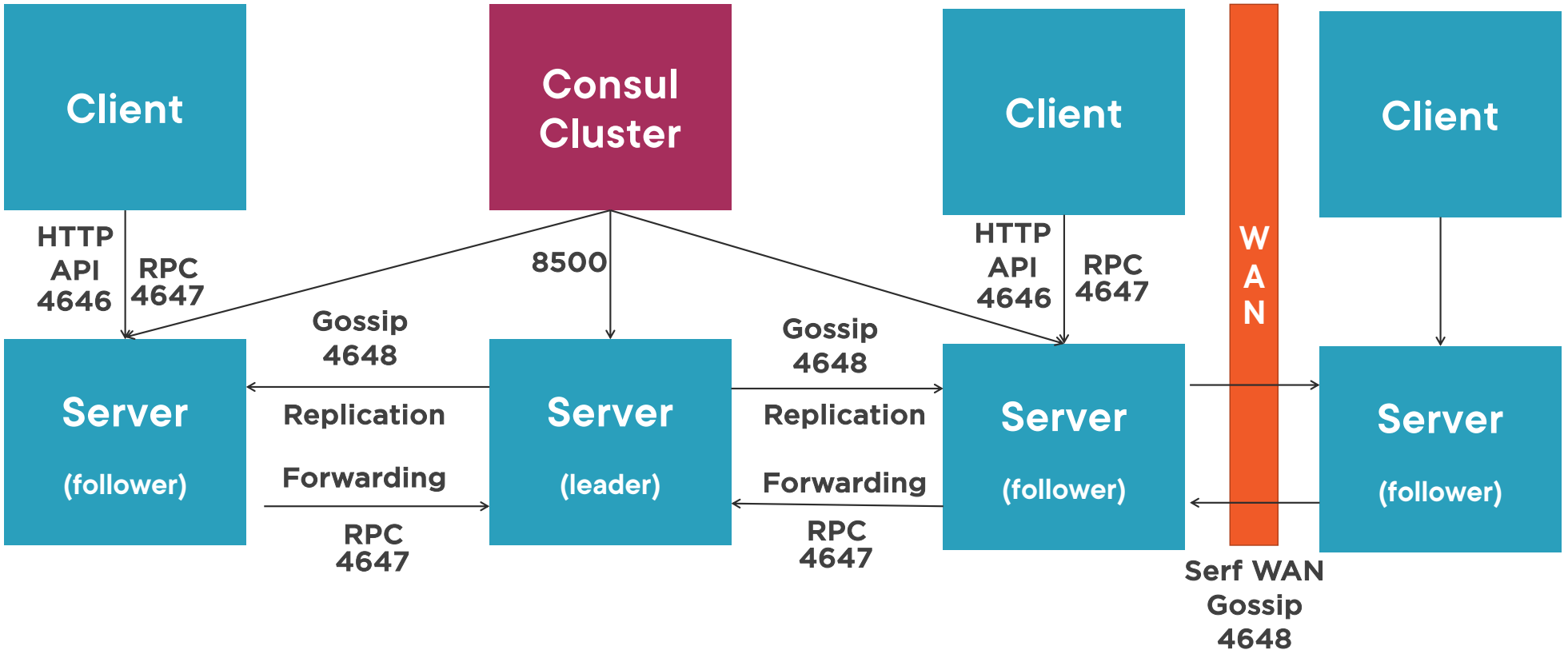
Nomad Host Volumes

Locally presented persistent storage

CSI Plugins

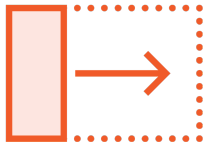
Uses CSI-certified backend storage

Networking Dependencies



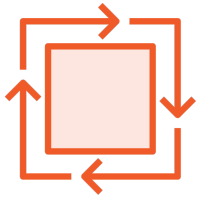
Nomad Deployment Patterns

Nomad Server Architectural Patterns



Nomad expects 3 or 5 servers in a cluster

Nomad servers expect <10ms latency



Nomad clients can be deployed across any environment

Nomad Server and Client Permissions

Server

Nomad servers should be run with the lowest possible permissions

Create a nomad user with the minimal required privileges

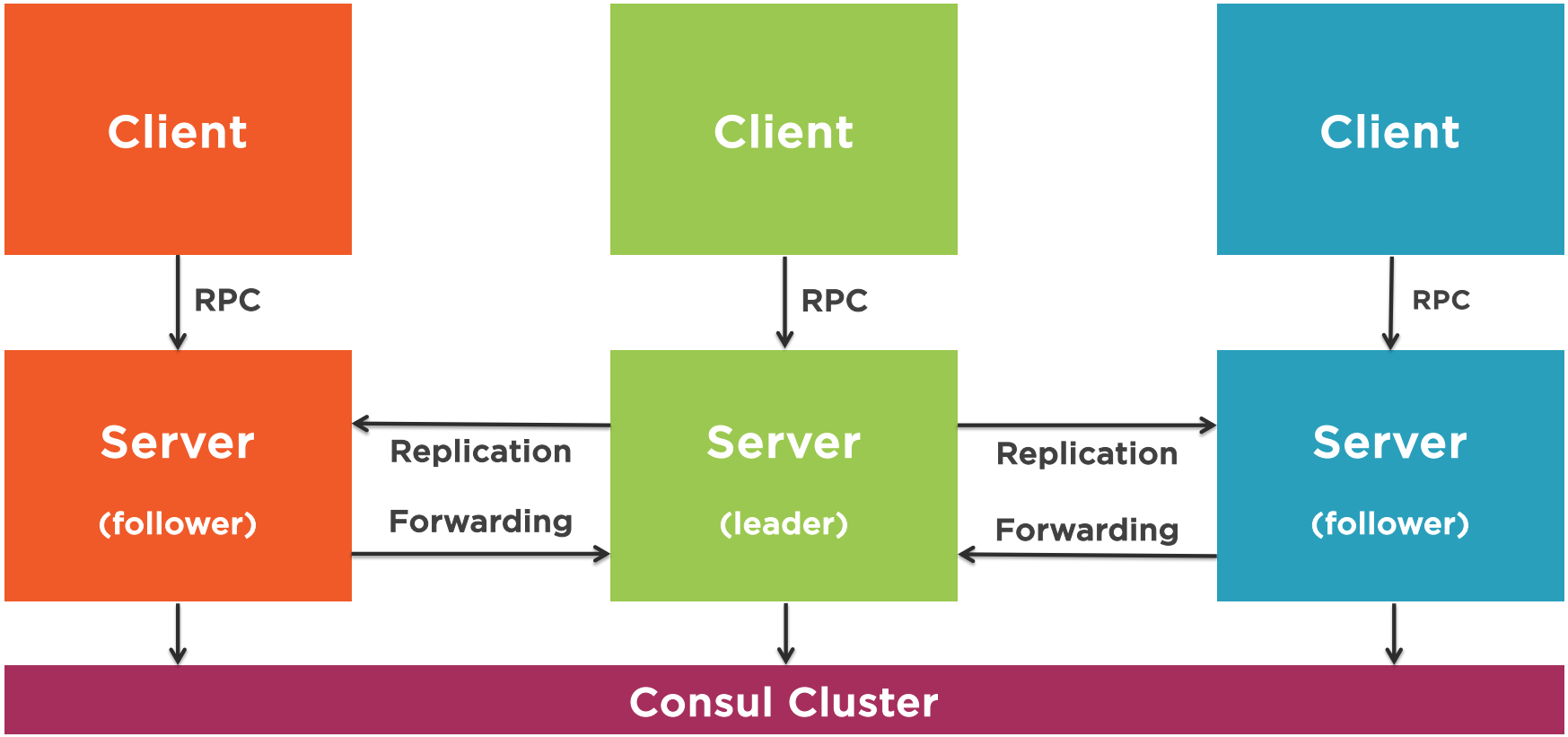
Client

Nomad clients must be run as root due to the OS isolation

Nomad client folder needs to be owned by root (0700 permission)

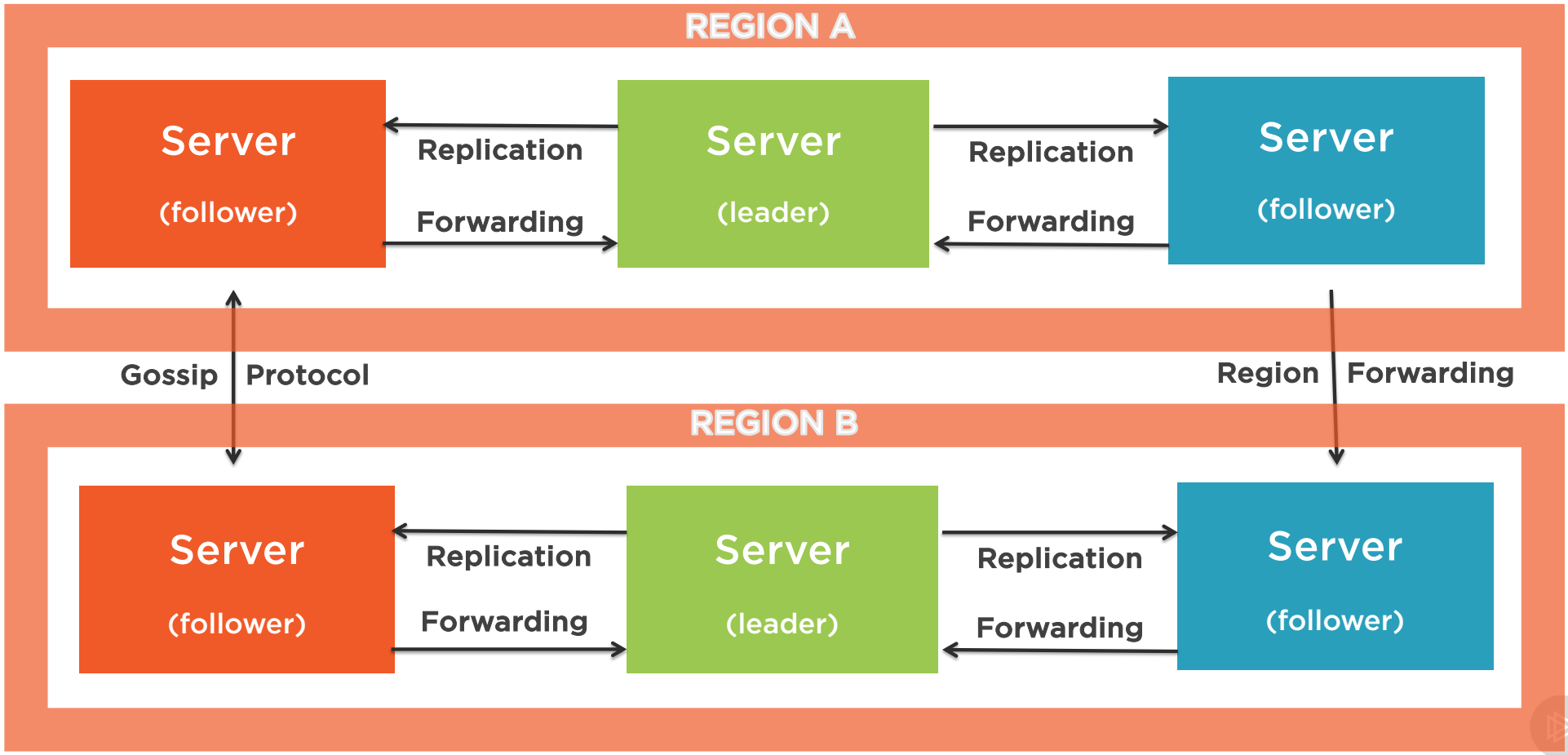
Single Region Deployment

Client/Server Deployment



Multi-Region Deployment

Multi-Region Deployment



Understanding Nomad Namespaces

Namespaces

Nomad also supports namespaces, which allows jobs and any of their associated objects to be segmented from each other and from other users of the cluster.

Nomad places all job-related objects including jobs, allocations, deployments, and evaluations, using namespaces.

Common Namespace Use-Cases

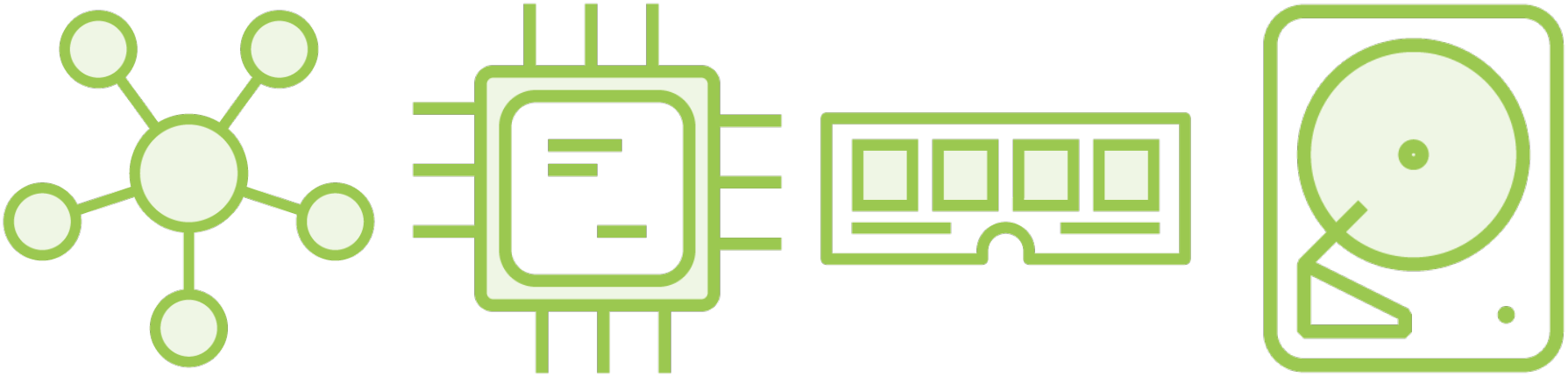
Segment between development teams

Segment environments within a cluster

Ensure co-location of resources for teams

Sizing your Nomad Deployment

What Resources Need Sizing?



# of Nodes

Autoscaling options available

Compute

Requires common processors

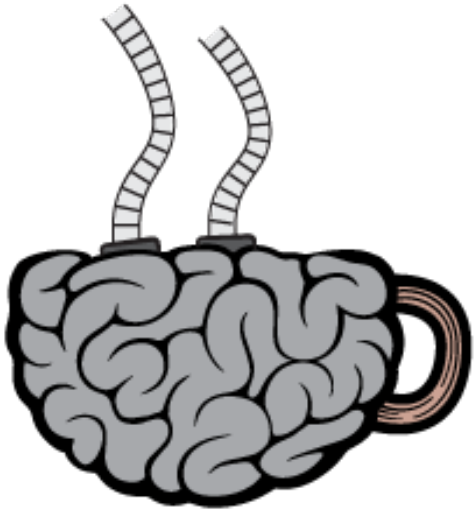
Memory

Use for rapid local caching

Storage

Ephemeral and persistent

Use-Case: Wired Brain Coffee



WIRED BRAIN

COFFEE

Application Re-Platform Project

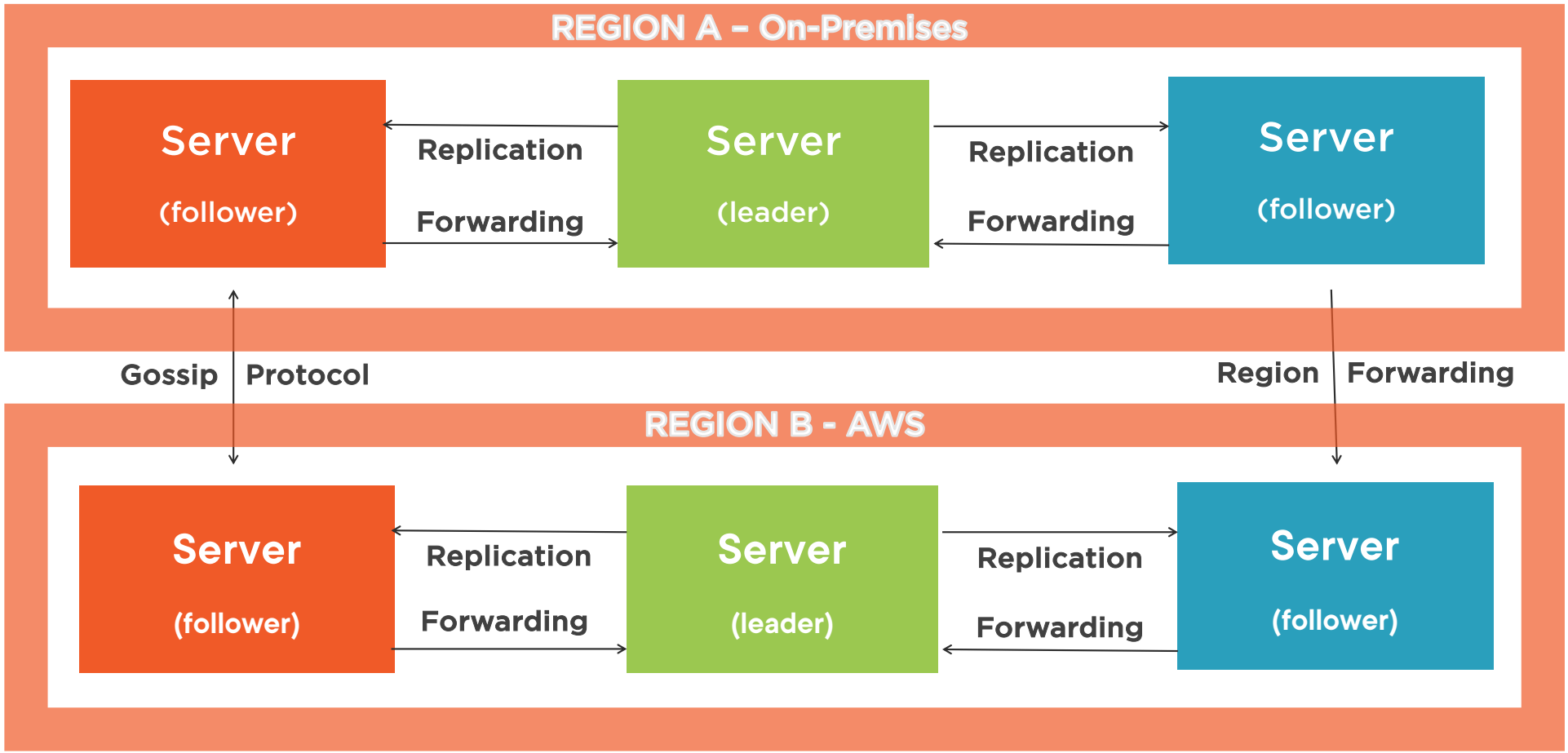
You are the Ops team lead at Wired Brain Coffee you have chosen Nomad as your hosting platform.

You are running a development environment and need to deploy production using Hybrid infrastructure.

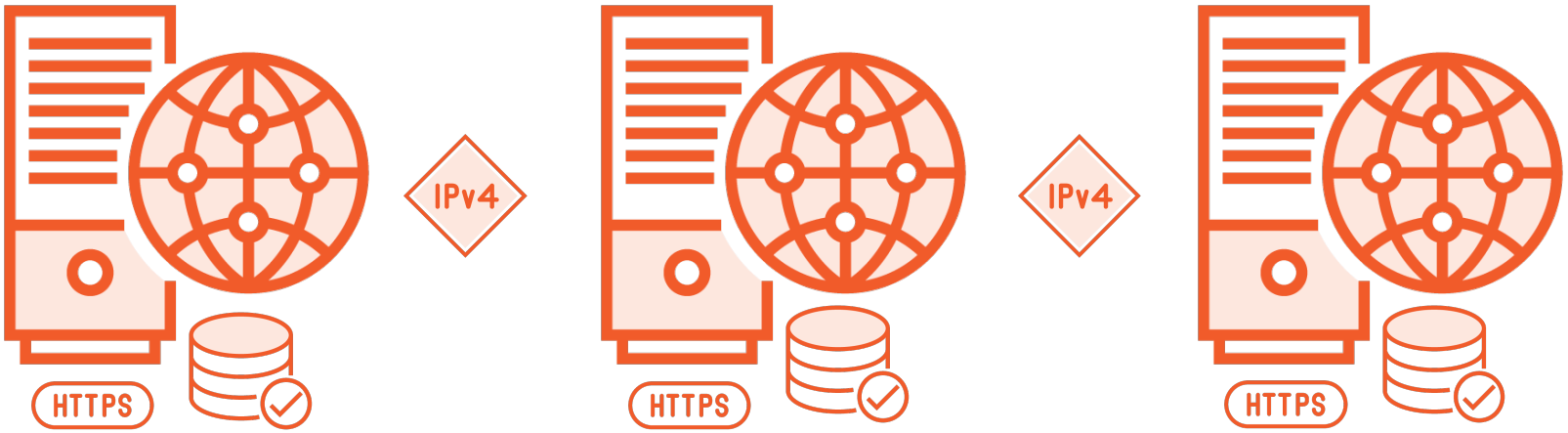
Wired Brain is using bare-metal on-premises and AWS in the cloud with a dedicated persistent VPN connection

The Wired Brain Hybrid Deployment

Wired Brain Coffee



Wired Brain Coffee – Local Data Center

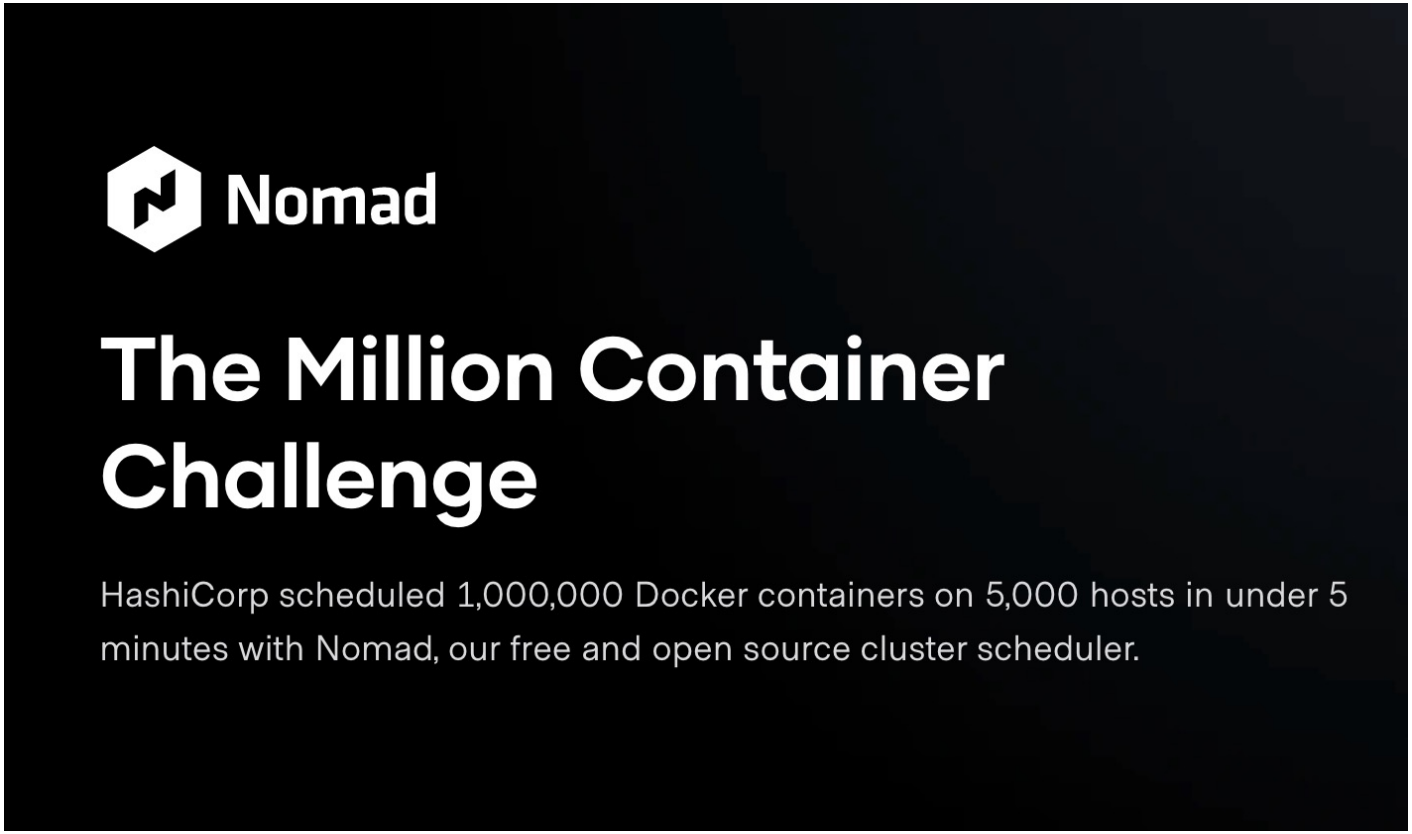


Node1
Nomad
Consul
Docker

Node2
Nomad
Consul
Docker

Node3
Nomad
Consul
Docker

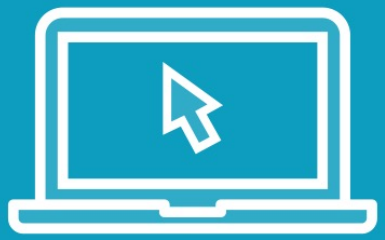
Nomad “At-Scale”



https://www.hashicorp.com/c1m

Nomad Deployment Using Infrastructure-as-Code

Demo



- **Review the code and configuration for a sample hybrid deployment of Nomad**

Summary

- Reviewed Nomad dependencies
- Explored deployment scenarios
- Chose the Wired Brain Coffee use-case
- Mapped out a hybrid Nomad deployment
- Looked at a code-based deployment of Nomad using Terraform and AWS

Up Next:
Deploying Nomad in a Hybrid Architecture